IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Craig William Fellenstein	§	Confirmation No.: 2482
	§	
Serial Number: 10/765,777	§	
	§	Group Art Unit: 2166
Filed: 01/27/2004	§	
	§	
For: SYSTEM AND METHOD FOR AUTONOMIC PERFORMANCE ENHANCEMENT OF STORAGE MEDIA	§	Examiner: Navneet K. Ahluwalia
	§	

Commissioner of Patents and Trademarks
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICANTS' APPEAL BRIEF

Applicant-inventors ("Applicants") and assignee International Business Machines Corporation respectfully submit the present brief in support of the patentability of the claims of the above-referenced application.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, of Armonk, New York, assignee of the interests in the invention from the named inventor.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

1, 3-5, 7-8, 10-11, and 13 are pending. Of these, Claims 1, 5, 8, 10, 11, and 13 are independent Claims. Claims 2, 6, 9, and 12 have been canceled. Applicants appeal the Examiner's rejections of Claims 1, 3-5, 7-8, 10-11, and 13 under 35 U.S.C. §103(a).

IV. STATUS OF AMENDMENTS

The Claims stand as amended in Applicants' Response to an Office Action dated July 1, 2008 ("Third Response").

V. SUMMARY OF CLAIMED SUBJECT MATTER

The Claims under examination are directed to improving the storage algorithms on storage media. *See* Application, Page 1, lines 9-10. Most storage media (*e.g.*, hard disk drives) are write-optimized. *See* Application, Page 1, lines 24-25. They are designed to get the data to the disk as efficiently as possible. *See* Application, Page 1, lines 25-27. In practice, this means that the drives will often break a file into fragments and use all available heads to write the fragments to different platters simultaneously. *See* Application, Page 1, line 27 to Page 2, line 1. This often results in files being broken into non-contiguous segments on disk, which causes diminished read performance. *See* Application, Page 2, lines 2-13. Periodic defragmentation helps reduce this problem. *See* Application, Page 2, lines 14-16. But typical periodic defragmentation is usually only done when the disk performance is so bad as to be noticeable.

See Application, Page 2, lines 18-20. Periodic defragmentation therefore causes problematic productivity delays. *See* Application, Page 2, lines 20-21.

The novel apparatuses, methods, and processors defined in Claims 1, 3-5, 7-8, 10-11, and 13 solve this problem by embodying a novel system and method for autonomic performance enhancement of storage media. Specifically, the claimed embodiments introduce a "fragmentation tracker" (FT) and a "defragmentation agent" (DA). *See* Application, Page 5, lines 19-24. Broadly, the FT monitors and maintains an account of the location of fragmented file clusters and the DA defragments the files identified by the FT. *See* Application, Page 5, line 24 to Page 6, line 2. More specifically, the FT monitors the storage media for scan/write/delete operations that cause file fragmentation, storing the location of fragmented data clusters and ignoring operations that do not cause fragmentation. *See* Application, Page 6, lines 6-18. The DA defragments those files identified by the FT, based on non-intrusive defragmentation operations, defragmenting files only when the system is idle. *See* Application, Page 7, lines 1-17.

This novel approach offers several significant advantages that are unavailable to the prior art methods or systems. First, this approach does not completely lock out disk usage once defragmentation begins. *See* Application, Page 7, lines 9-10. Second, the DA operates whenever the system is idle, instead of waiting until fragmentation causes substantial or critical delays. *See* Application, Page 6, lines 22-24. Third, the DA does not require a user to initiate defragmentation. *See* Application, Page 6, lines 20-21. Other benefits are enumerated throughout the Application.

The Claims are as follows, shown with illustrative citations to page and line numbers in the Original Application designated in curved braces ("{}"):

1. (Previously Presented) An apparatus for file defragmentation of at least one storage medium, comprising:

a computer system at least coupled to the at least one storage medium; {Page 5, Lines 1-5}

a tracker, wherein the tracker is at least configured to maintain a record of at least locations of a plurality of file fragments on at least one storage medium; {Page 5, Line 25 – Page 6, Line 2}and

an agent, {Page 6, Line 2}wherein the agent is at least:

configured to operate while the computer system is at least idle; {Page 7, Lines 1-5}

configured to defragment the plurality of file fragments; {Page 7, Lines 4-5}

configured to modify attributes of defragmentation; and {Page 8, Lines 4-16}

configured to delete the record of at least locations of the plurality of file fragments {Page 7, Lines 7-8}.

5. (Previously Presented) An apparatus for file defragmentation of at least one storage medium at least coupled to a computer system, comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments; {Page 5, Lines 1-5}

an idle monitor, wherein the idle monitor is at least configured to enable defragmentation while the computer system is at least idle; {Page 6, Lines 18-19}

a defragmenter, wherein the defragmenter is at least configured to defragment the plurality of file fragments and to modify attributes of defragmentation; {Page 6, Line 2; Page 9, Lines 17-24} and

an update monitor, wherein the update monitor is at least configured to delete a record in the memory of at least locations of the plurality of file fragments that at least been defragmented {Page 7, Lines 7-8}.

8. (Original) A method of for file defragmentation of at least one storage medium coupled to a computer system, comprising:

determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media; {Page 5, Line 25 – Page 6, Line 2}

storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium; {Page 6, Lines 10-14}

determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, defragmenting a file; {Page 7, Lines 4-5}

determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, determining if defragmentation is stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, reporting an error. {Page 7, Lines 16-17}

10. (Original) A method of defragmenting at least one storage medium coupled to a computer system, comprising:

determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, defragmenting the file; {Page 7, Lines 4-5}

determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, deleting a location of the fragmented file clusters in a storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, determining if stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, reporting an error. {Page 7, Lines 16-17}

11. (Previously Presented) A processor comprising a computer program for file defragmentation of at least one storage medium at least coupled to a computer system, the computer program embodied on a tangible computer-readable medium and comprising:

computer code for determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media; {Page 5, Line 25 – Page 6, Line 2}

computer code for storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium; {Page 6, Lines 10-14}

computer code for determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, computer code for sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, computer code for defragmenting a file; {Page 7, Lines 4-5}

computer code for determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, computer code for deleting the location of the fragmented file clusters in the storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, computer code for determining if defragmentation is stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, computer code for sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, computer code for reporting an error. {Page 7, Lines 16-17}

13. (Previously Presented) A processor comprising a computer program for defragmenting at least one storage medium coupled to a computer system, the computer program embodied on a tangible computer-readable medium and comprising:

computer code for determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, computer code for sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, computer code for defragmenting a file; {Page 7, Lines 4-5}

computer code for determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, computer code for deleting a location of the fragmented file clusters in a storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, computer code for determining if stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, computer code for sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, computer code for reporting an error. {Page 7, Lines 16-17}

VI. GROUNDS OF REJECTION TO BE REVIEWED

Whether Claims 1, 3-5, 7-8, 10-11, and 13 are patentable over Jochemsen et al. (US 6,757,804 B2)("Jochemsen") in view of Carlson (US 2003/0101383 A1)("Carlson") under 35 U.S.C. §103(a).

VII. ARGUMENT

A. Grouping of Claims

Claims 1, 5, 8, 10, 11, and 13 are independent. For purposes of this appeal, Applicants consider each of the independent Claims, and their respective dependent Claims, as separate groups. Thus, the groups of Claims are (1, 3-4), (5, 7), 8, 10, 11, and 13.

B. Summary of Pertinent Prosecution

Applicants filed the present application on January 27, 2004, with 13 claims.

The Examiner mailed the First Office Action on September 6, 2006 ("First Action"), rejecting Claims 1-13 under 35 U.S.C. §102(e) as allegedly anticipated by Jochemsen.

Applicants responded to the first Office Action on December 6, 2006 ("First Response"), amending Claim 4 to depend from Claim 1 and cancelling Claims 9 and 12. The First Response also argued that Jochemsen failed to teach every element of the Claims. Specifically, Applicants noted that Jochemsen does not teach file defragmentation or operation when the system is idle. *See* First Response, Pages 8-9.

The Examiner mailed a Final Action on March 7, 2007 ("First Final Action"), maintaining the rejections in the First Action. During a telephone interview with the Examiner and the Examiner's supervisor, conducted on May 16, 2007 ("First Interview"), Applicants' representative enumerated additional limitations not shown in Jochemsen, particularly the steps of Claim 8, including "determining if defragmentation is complete," "if defragmentation is not complete, determining if defragmentation is stopped by activity," "determining if the computer system is idle," and others. The Examiner requested Applicants to provide the discussed argument in a written response after final.

As requested, Applicants responded to the Final Action on June 7, 2007 ("Second Response") detailing the shortcomings of the reference as prior art under Section 102, as discussed in the Interview, and reiterating the arguments presented in the First Response.

Nevertheless, on June 26, 2007, the Examiner mailed an Advisory Action maintaining the rejections of Claims 1-8, 10, 11, and 13, without addressing the numerous missing elements in the reference. Applicants appealed and filed an Appeal Brief on January 31, 2007 ("First Appeal Brief").

In view of the First Appeal Brief, the Examiner reopened prosecution in a non-final Office Action mailed January 8, 2008 ("Second Action"). In the Second Action, the Examiner rejected the Claims under 35 U.S.C. §103(a) as allegedly unpatentable over Jochemsen in view of Carlson. The Examiner also rejected Claims 11 and 13 under 35 U.S.C. §101 and §112, second paragraph.

Applicants responded on July 1, 2008 ("Third Response"). In the Third Response, Applicants argued that the Examiner failed to establish a *prima facie* case of obviousness as the Examiner's proposed combination of Jochemsen and Carlson failed to show or even suggest

several missing Claim elements. In the Third Response, Applicants amended Claims 1 and 5 to include an element from a dependent Claim. Applicants also amended Claims 11 and 13 to overcome the rejections under 35 U.S.C. §101 and §112, second paragraph.

The Examiner mailed the Final Action under appeal on October 6, 2008 ("Second Final Action"). In the Second Final Action, the Examiner maintained the Section 103 rejections of the Second Action.

This appeal followed.

The Examiner's Rejections Were Procedurally and Factually in Error

1. The Form and Content of the Examiner's Rejections under Section 103 Were Improper and Insufficient

a. Legal Requirements for an Obviousness Rejection

The M.P.E.P. clearly defines the obligation of the examiner to produce reasoning and evidence in support of obviousness:

> The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of nonobviousness.
>
> M.P.E.P. §2142.

To make a *prima facie* case, the Examiner must perform certain established inquiries: "Obviousness is a question of law based on underlying factual inquiries." M.P.E.P. §2141(II). These factual inquiries include determining the scope and content of the prior art, "Ascertaining the differences between the claimed invention and the prior art;" and "Resolving the level of ordinary skill in the pertinent art." M.P.E.P. §2141(II)(*citing* Graham v. John Deere Co., 383 U.S. 1, at 17-18, 148 USPQ 459, at 467 (1966)).

These factual inquiries required to establish a *prima facie* case are not optional: "Office personnel fulfill the critical role of factfinder when resolving the Graham inquiries . . . When making an obviousness rejection, Office personnel must therefore ensure that the written record includes findings of fact concerning the state of the art and the teachings of the references applied." M.P.E.P. §2141(II)(emphasis added). Thus, the Examiner *must* make these three factual inquiries in order to support a proper Section 103 rejection. Without these factual inquiries, the Section 103 rejection cannot stand: "Factual findings made by Office personnel are the necessary underpinnings to establish obviousness." M.P.E.P. §2141(II).

Furthermore, in undertaking the mandatory factual inquiries, the Examiner must contemplate the Claims as a whole: "In view of all factual information, the examiner must then make a determination whether the claimed invention 'as a whole' would have been obvious at that time to that person." M.P.E.P. §2141. Moreover, the Examiner must consider each and every element in the pending Claims: "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970); M.P.E.P. §2143.03.

As stated in the MPEP, "The key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious." M.P.E.P. §2142. The Examiner's rejections in this case fail to ascertain all of the differences between the claimed invention and the prior art and fail to show how those differences would have been obvious. "[R]ejections on obviousness cannot be sustained with mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329,

1336 (Fed. Cir. 2006); M.P.E.P. §2142. Applicants respectfully submit that the Examiner has not met this burden and has therefore failed to establish a *prima facie* case of obviousness.

b. The Examiner has failed to establish a *prima facie* case of obviousness

As described above, the Examiner rejects Claims under 1, 3-5, 7-8, 10-11, and 13 under 35 U.S.C. §103(a) as allegedly unpatentable over Jochemson in view of Carlson. *See* Second Final Action, Page 3. Applicants respectfully submit that these rejections are in error and should be withdrawn.

More particularly, Applicants respectfully submit that the Examiner has failed to establish *prima facie* obviousness for each Claim because the Examiner has at least failed to produce evidence showing the teaching or suggestion of all the Claim limitations by the references here.

First, regarding Claim 8, the Examiner asserts that Jochemsen teaches "determining if the computer system is idle if the computer system is not idle, sleeping for an interval." Second Final Action, Page 7 (*citing* Jochemsen, col. 1, lines 49-58). Nowhere does Jochemsen disclose "determining if the computer system is idle" or performing any actions based on that determination. The Examiner's citation does not even hint at "determining if the computer system is idle":

> When a delete operation is necessary, it is desirable permanently to delete the minimum number of files so that (a) at least the required amount of free space is created; and (b) defragmentation and free-space defragmentation are optimized.
>
> It is an aim of preferred embodiments of the present invention to provide a method, system and corresponding computer program product for reducing fragmentation of a digital storage device.

Jochemsen, col. 1, lines 49-58. As shown, the cited passage does not come anywhere close to teaching any determination of whether the computer system is idle. Applicants

respectfully submit that nowhere else does Jochemsen even hint at "determining if the computer system is idle."

Instead, and contradicting the Examiner's earlier assertion, the Examiner states "Jochemsen does not explicitly disclose as it is silent about the system operating when idle." Second Final Action, Page 7. Applicants respectfully note that "the system operating when idle" does not imply or suggest "determining if the computer system is idle," as recited in the Claim. Nevertheless, the Examiner offers Carlson as purportedly teaching "the system working at all times and explicitly discloses about the idle system." Second Final Action, Page 7 (*citing* Carlson, paragraphs [0026]-[0027]). But neither does the Carlson reference, in the cited passage or elsewhere, teach or suggest "determining if the computer system is idle," as recited in the Claim.

Instead, Carlson teaches "the file system subsystem determines whether more storage device I/O requests are pending in [I/O] queue 112." Carlson, para. [0027]. "If the I/O queue is empty, meaning that the *storage device* 108 would be idle anyway, then the file system subsystem determines that file maintenance can occur." Carlson, para. [0027] (emphasis added). Applicants respectfully note that Carlson plainly teaches determining whether an I/O queue is empty, on the explicit assumption that the empty Carlson I/O queue is a sufficient condition to identify an idle *storage device*. Thus, Carlson expressly states that it is not determining whether "the computer system is idle," but rather whether a queue is empty.

Nor does the Examiner assert that Carlson teaches actually "determining whether the computer system is idle." Instead, the Examiner offers Carlson as showing, "the system working at all times and explicitly discloses *about the idle system.*" Second Final Action, Page 7 (*citing* Carlson, paragraphs [0026]-[0027])(emphasis added). As described above, what Carlson

discloses "about the idle system" is that Carlson determines whether a queue is empty, not, as recited in the Claim "whether the computer system is idle." As such, Applicants respectfully submit that the Examiner's own argument defeats any *prima facie* case of obviousness here.

Accordingly, Applicants are under no obligation to offer any further evidence or reasoning establishing the nonobviousness of Claim 8. *See* M.P.E.P. §2142. Nevertheless, Applicants respectfully note that an empty instruction queue for a particular storage device is not an adequate determination of whether a computer system is "idle," and does not render obvious a determination of whether that computer system is "idle." For example, in systems with multiple storage devices, a computer system may be busy (*i.e.*, not idle) transferring data to a system memory from a storage device that does not have an empty I/O queue. *See*, *e.g.*, Carlson, para. [0035]. In such a case, one "storage device" may be idle while the "computer system" is not idle. Thus, even if the Examiner asserted Carlson as showing something more than "about the idle system," Applicants respectfully submit that Carlson fails to teach or suggest "determining if the computer system is idle" as recited in Claim 8.

And "determining if the computer system is idle" is not the only element missing from the Examiner's proposed combination. For example, nowhere does Jochemsen/Carlson teach "determining if defragmentation is complete [sic] if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium" as the Examiner alleges. Second Final Action, Page 7 (*citing* Jochemsen, col. 1, lines 49-53). As shown above, the cited reference does not teach anything close to "determining if defragmentation is complete." Nor does the Examiner assert Carlson as teaching this element.

Similarly, Jochemsen/Carlson completely fails to teach, "if defragmentation is not complete, determining if defragmentation is stopped by activity" as the Examiner alleges.

Second Final Action, Page 7 (*citing* Jochemsen, col. 3, lines 22-34). The wholly inapposite cited passage states:

> When a deletion operation is initiated (step 200) the file manager 30 is interrogated (step 202) to determine whether multiple files are available for deletion. A file is available for deletion if it has been so annotated. A file is only available for deletion if in doing so it provides sufficient free-space for the required write operation. This may mean that several of a plurality of files need to be deleted. If only a single file is available for deletion, it is deleted (step 204) and the file manager 30 updated accordingly (step 210). If multiple files are available a file is selected to be deleted to reduce fragmentation (step 206). Options for selecting the file to be deleted in this embodiment of the present invention are set out below. Next the selected file is permanently deleted (step 208) and the file manager 30 is updated accordingly (step 210).

Jochemsen, col. 3, lines 22-36. As shown, nowhere in the cited passage does Jochemsen even hint at "stopping defragmentation" much less "determining whether defragmentation is stopped by activity," as recited in Claim 8. Applicants respectfully submit that nowhere else does Jochemsen or Carlson even hint at "determining whether defragmentation is stopped by activity." Nor does the Examiner anywhere suggest that Carlson supplies this missing element. Accordingly, the Examiner's proposed combination of Jochemsen/Carlson fails to show this element as well.

Likewise, Jochemsen/Carlson also fails to show, among other things, "sleeping for an interval" or "reporting an error" as a function of "determining if defragmentation is stopped by activity." Nor does the Examiner provide any evidence of these missing elements. The Examiner does not even suggest that Carlson provides these missing elements and merely cites various inapposite passages of Jochemsen. *See* Second Final Action, Page 7. For example, to support "if defragmentation is stopped by activity, sleeping for an interval" the Examiner cites "column 2, lines 1-10, Jochemsen". Second Final Action, Page 7. But the cited passage does not

say anything even remotely relevant to "if defragmentation is stopped by activity, sleeping for an interval":

> According to the present invention in a second aspect, there is provided a system arranged for reducing fragmentation of a digital storage device, characterized by the system comprising means for determining that a plurality of files is available for deletion; means for selecting one of the files; and means for deleting the selected file and not deleting another of the files.
>
> It has been realized that with digital storage devices annotating a plurality of files for deletion, there is an efficient opportunity to reduce fragmentation by selective file deletion.

Jochemsen, col. 2, lines 1-11. As stated above, "[R]ejections on obviousness cannot be sustained with mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006); M.P.E.P. §2142. Applicants respectfully submit that a bald citation to an irrelevant passage, without even accompanying explanation, comes nowhere close to "articulated reasoning with some rational underpinning."

Thus, for at least the above reasons, Applicants have shown that the Examiner's proposed combination of Jochemsen/Carlson cannot support *prima facie* obviousness or a Section 103 rejection of Claim 8, as Jochemsen/Carlson completely fails to teach or suggest several elements of Claim 8. Accordingly, Applicants respectfully assert that the rejection of Claim 8 is in error and should be withdrawn.

Next, Applicants respectfully submit that Jochemsen/Carlson also fails to teach each and every element of the remaining pending Claims, in a similarly clear manner as above. Specifically, the Examiner's cited passages in Jochemsen wholly fail to even mention, much less support, the elements they are offered to support, while the secondary reference likewise fails to teach these elements. For example, regarding Claim 10, Jochemsen fails to teach at least the

following elements recited in Claim 10, "determining if the computer system is idle", "determining if defragmentation is complete", "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or "reporting an error" based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen.

Regarding Claim 10, the Examiner cites a different passage to support "determining if the computer system is idle," than the passage offered against Claim 8, but, as noted above, that passage also teaches nothing anywhere close to "determining if the computer system is idle." *See* Second Final Action, Page 8 (*citing* Jochemsen, col. 2, lines 1-10). Applicants respectfully submit that nowhere else does Jochemsen or Carlson teach "determining if the computer system is idle," as recited in Claim 10.

Instead, and again contradicting the Examiner's own assertion, the Examiner offers the same reasoning as applied to Claim 8, stating, "Jochemsen does not explicitly disclose as it is silent about the system operating when idle." Second Final Action, Page 8. Applicants again respectfully note that "the system operating when idle" does not imply or suggest "determining if the computer system is idle," as recited in the Claim. As before, the Examiner offers Carlson as purportedly teaching "the system working at all times and explicitly discloses about the idle system." Second Final Action, Page 8 (*citing* Carlson, paragraphs [0026]-[0027]). But neither does the Carlson reference, in the cited passage or elsewhere, teach or suggest "determining if the computer system is idle," as recited in the Claim, as shown above.

Thus, for at least the above reasons, Applicants have shown that the Examiner's proposed combination of Jochemsen/Carlson cannot support *prima facie* obviousness or a Section 103 rejection of Claim 10, as Jochemsen/Carlson completely fails to teach several elements of Claim

10. Accordingly, Applicants respectfully assert that the rejection of Claim 10 is in error and should be withdrawn.

Next, regarding Claim 11, Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson also fails to teach at least the following elements recited in Claim 11: "determining if the computer system is idle", "determining if defragmentation is complete," "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or "reporting an error" based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen/Carlson.

As above, and again contradicting the Examiner's own citation, the Examiner offers the same reasoning as applied to Claim 8 and Claim 10, stating, "Jochemsen does not explicitly disclose as it is silent about the system operating when idle." Second Final Action, Page 10. The Examiner offers Carlson as purportedly teaching "the system working at all times and explicitly discloses about the idle system." Second Final Action, Page 10 (*citing* Carlson, paragraphs [0026]-[0027]). Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson cannot support *prima facie* obviousness or a Section 103 rejection of Claim 11 for at least the same reasons why the *prima facie* case fails against Claims 8 and 10. Accordingly, Applicants respectfully submit that the rejection of Claim 11 is in error and should be withdrawn.

Next, regarding Claim 13, Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson fails to teach at least the following elements recited in Claim 13: "determining if the computer system is idle", "determining if defragmentation is complete," "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or

"reporting an error" based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen/Carlson.

As above, and again contradicting the Examiner's own citation, the Examiner offers the same reasoning as applied to Claims 8, 10 and 11, stating, "Jochemsen does not explicitly disclose as it is silent about the system operating when idle." Second Final Action, Page 11. The Examiner offers Carlson as purportedly teaching "the system working at all times and explicitly discloses about the idle system." Second Final Action, Page 11 (*citing* Carlson, paragraphs [0026]-[0027]). Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson cannot support *prima facie* obviousness or a Section 103 rejection of Claim 13 for at least the same reasons why the *prima facie* case fails against Claims 8, 10 and 11. Accordingly, Applicants respectfully submit that the rejection of Claim 13 is in error and should be withdrawn.

Next, regarding Claims 1 and 5, Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson similarly fails to teach the ability to "modify attributes of defragmentation," as asserted by the Examiner. Second Final Action, Pages 3 and 5 (*citing* Jochemsen, col. 3, lines 37-46). Applicants respectfully submit that nowhere does Jochemsen or Carlson, either in the cited passages or elsewhere, recite the above limitation. Instead, the cited passage teaches selecting one of a number of files for deletion, stating, "there can be several criteria to determine the text file to delete." Jochemsen, col. 3, lines 40-41. Applicants respectfully submit that this fails to teach the ability to "modify attributes of defragmentation," as recited in Claims 1 and 5.

Moreover, the Examiner cites Jochemsen as teaching, "an agent, wherein the agent is at least: configured to operate while the computer system is at least idle (column 2 lines 58-67,

Jochemsen); configured to defragment the plurality of file fragments (column 2, lines 1-10, Jochemsen); and configured to delete the record of at least locations of the plurality of file fragments (column 2 lines 58-67 and column 3 lines 23-45)." Second Final Action, Page 3.

The Examiner cites column 2, lines 1-10, of Jochemsen as teaching defragmentation. However, as described above, the actual text instead teaches "reducing fragmentation" not active defragmentation. Applicants respectfully submit that Jochemsen teaches deletion and writing in a manner to reduce fragmentation of a file *before* the file is written to a disk. Conventional defragmentation takes place *after* a series of files have been written to a disk. In Jochemsen, no individual clusters are moved or rearranged to form more contiguous units. Instead, Jochemsen is essentially a specialized delete function in which files to be deleted are analyzed along with the fragmented free space and the size of a write file in order to delete only those files necessary to place the write file on the disk in a manner that reduces the overall fragmentation of the disk. *See* Jochemsen, col. 2, lines 51-58.

As above, the Examiner offers against Claims 1 and 5 the same reasoning as applied to Claims 8, 10, 11 and 13, stating "Jochemsen does not explicitly disclose as it is silent about the system operating when idle." Second Final Action, Pages 3 and 5. The Examiner offers Carlson as purportedly teaching "the system working at all times and explicitly discloses about the idle system." Second Final Action, Pages 3 and 5 (*citing* Carlson, paragraphs [0026]-[0027]). Applicants respectfully submit that the Examiner's proposed combination of Jochemsen/Carlson cannot support *prima facie* obviousness or a Section 103 rejection of Claim 1 or 5 for at least the same reasons why the *prima facie* case fails against Claims 8, 10, 11 and 13. Accordingly, Applicants respectfully submit that the rejections of Claims 1 and 5 are in error and should be withdrawn.

The Examiner attempts to rebut the above arguments by restating the rejections:

> Jochemsen in combination with Carlson teaches the determination of the system being idle and also defragmentation taking place in Carlson of the fragmentation occurring even when system is idle enhances the productivity and execution of the system paragraphs 26-27. Furthermore, in column 2 lines 58-67 and column 3 lines 23-45, Jochemsen discloses the deletion of records and the functioning of determination of system when its idle. [sic]

Second Final Action, Page 2. As described above, Applicants have overcome this restatement of the Examiner's rejections

Next, regarding Claims 3 and 7, the Examiner's proposed combination of Jochemsen/Carlson wholly fails to even mention the missing elements. Applicants respectfully submit that neither Jochemsen nor Carlson teach modifying the attributes of defragmentation "wherein the attributes are selected from the group consisting of *file type*, frequency of access, *typical access duration*, interval between accesses, file/application association, file size, *read attributes*, *update attributes*, and time of day of typical access," as asserted by the Examiner. Second Final Action, Pages 4 and 6 (*citing* Jochemsen, col. 4, lines 22-226 [sic], and 42-47)(emphasis added). While the Examiner relies exclusively on Jochemsen as purportedly showing the above elements, Applicants respectfully submit that Carlson also fails to teach these elements. Applicants respectfully submit that nowhere does Jochemsen, either in the cited passages or elsewhere, recite the above limitation.

For example, Jochemsen recites:

> The determination of which file to delete whichever selection option is chosen is carried out by interrogating the file manager 30. For every file in a set of deletable files the effect of deletion on file fragmentation and free-space fragmentation is calculated.
>
> . . .
>
> The effect on free-space fragmentation is calculated by calculating the changes the deletion (for each file) would have on the free-space fragments (step 402).

> New free-space fragments might appear (undesirable), but several free-space fragments can be connected by deletion creating new free-space fragments (desirable). In general, the number of free-space fragments will increase (i.e. a positive change).

Jochemsen, col. 4, lines 22-26, 42-47. Clearly, this passage does not teach "wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access," as recited in Claims 3 and 7. Applicants respectfully submit that nowhere else does Jochemsen or Carlson teach or even suggest this element.

Applicants have now positively demonstrated the pervasive failure of the Examiner's references to teach or suggest the elements recited in the Claims. As stated above, the Examiner must consider each and every element in the pending Claims: "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970); M.P.E.P. §2143.03. In this case, the Examiner has failed to consider "all words" in the Claims, as evidenced by the repeated citations to passages that are wholly irrelevant to the words used in the Claims, and the failure to bring references teaching or suggesting multiple elements. As such, the Examiner has failed to bring a proper *prima facie* case of obviousness. Therefore, Applicants respectfully submit that the Examiner's rejections are in error and should be withdrawn.

As the Examiner's only rejections of the Claims are under Section 103, Applicants respectfully submit that the Claims now stand in full condition for allowance. Applicants therefore respectfully request full allowance of Claims 1, 3-5, 7-8, 10-11, and 13.

VIII. CLAIMS APPENDIX

See Attached.

IX. EVIDENCE APPENDIX

NONE.

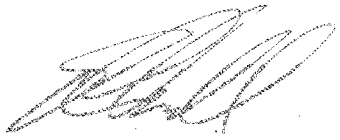
X. RELATED PROCCEDINGS APPENDIX

NONE.

XI. CONCLUSION

For the foregoing reasons, it is respectfully submitted that the Final Rejections of Claims 1, 3-5, 7-8, 10-11, and 13 under 35 U.S.C. §103(a) are improper and should be reversed. Applicants respectfully request that the rejections of Claims 1, 3-5, 7-8, 10-11, and 13 be withdrawn and that Claims 1, 3-5, 7-8, 10-11, and 13 be allowed.

Respectfully submitted,



Dated: 6 Mar 09

Patrick E. Caldwell, Esq.
Reg. No. 44,580
The Caldwell Firm, LLC
PO Box 59655
Dallas, Texas 75229-0655
Telephone: (972) 243-4523

VIII – APPENDIX – CLAIMS ON APPEAL

1. (Previously Presented) An apparatus for file defragmentation of at least one storage medium, comprising:

a computer system at least coupled to the at least one storage medium;

a tracker, wherein the tracker is at least configured to maintain a record of at least locations of a plurality of file fragments on at least one storage medium; and

an agent, wherein the agent is at least:

configured to operate while the computer system is at least idle;

configured to defragment the plurality of file fragments;

configured to modify attributes of defragmentation; and

configured to delete the record of at least locations of the plurality of file fragments.

2. (Cancelled).

3. (Previously Presented) The apparatus of Claim 1, wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access.

4. (Previously Presented) The apparatus of Claim 1 further comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments;

a system monitor, wherein the system monitor at least determines if file fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage medium; and

an accounting means, wherein the accounting means is at least configured to store locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred.

5. (Previously Presented) An apparatus for file defragmentation of at least one storage medium at least coupled to a computer system, comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments;

an idle monitor, wherein the idle monitor is at least configured to enable defragmentation while the computer system is at least idle;

a defragmenter, wherein the defragmenter is at least configured to defragment the plurality of file fragments and to modify attributes of defragmentation; and

an update monitor, wherein the update monitor is at least configured to delete a record in the memory of at least locations of the plurality of file fragments that at least been defragmented.

6. (Cancelled).

7. (Previously Presented) The apparatus of Claim 5, wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access.

8. (Original) A method of for file defragmentation of at least one storage medium coupled to a computer system, comprising:

determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media;

storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium;

determining if the computer system is idle;

if the computer system is not idle, sleeping for an interval;

if the computer system is idle, defragmenting a file;

determining if defragmentation is complete;

if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium;

if defragmentation is not complete, determining if defragmentation is stopped by activity;

if defragmentation is stopped by activity, sleeping for an interval; and

if defragmentation is not stopped by activity, reporting an error.

9. (Canceled).

10. (Original) A method of defragmenting at least one storage medium coupled to a computer system, comprising:

determining if the computer system is idle;

if the computer system is not idle, sleeping for an interval;

if the computer system is idle, defragmenting the file;

determining if defragmentation is complete;

if defragmentation is complete, deleting a location of the fragmented file clusters in a storage medium;

if defragmentation is not complete, determining if stopped by activity;

if defragmentation is stopped by activity, sleeping for an interval; and

if defragmentation is not stopped by activity, reporting an error.

11. (Previously Presented) A processor comprising a computer program for file defragmentation of at least one storage medium at least coupled to a computer system, the computer program embodied on a tangible computer-readable medium and comprising:

computer code for determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media;

computer code for storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium;

computer code for determining if the computer system is idle;

if the computer system is not idle, computer code for sleeping for an interval;

if the computer system is idle, computer code for defragmenting a file;

computer code for determining if defragmentation is complete;

if defragmentation is complete, computer code for deleting the location of the fragmented file clusters in the storage medium;

if defragmentation is not complete, computer code for determining if defragmentation is stopped by activity;

if defragmentation is stopped by activity, computer code for sleeping for an interval; and

if defragmentation is not stopped by activity, computer code for reporting an error.

12. (Canceled).

13. (Currently Amended) A processor comprising a computer program for defragmenting at least one storage medium coupled to a computer system, the computer program embodied on a tangible computer-readable medium and comprising:

computer code for determining if the computer system is idle;

if the computer system is not idle, computer code for sleeping for an interval;

if the computer system is idle, computer code for defragmenting a file;

computer code for determining if defragmentation is complete;

if defragmentation is complete, computer code for deleting a location of the fragmented file clusters in a storage medium;

if defragmentation is not complete, computer code for determining if stopped by activity;

if defragmentation is stopped by activity, computer code for sleeping for an interval; and

if defragmentation is not stopped by activity, computer code for reporting an error.